

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-110768

(43)Date of publication of application : 25.04.1995

(51)Int.Cl.

G06F 9/38

G06F 9/45

(21)Application number : 05-255497

(71)Applicant : HITACHI LTD

(22)Date of filing : 13.10.1993

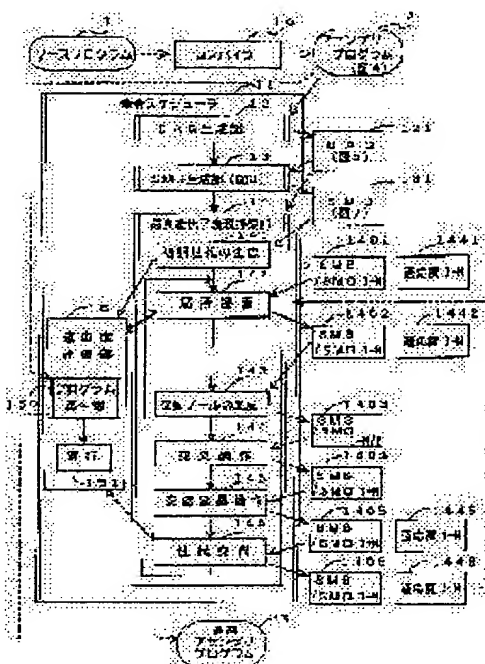
(72)Inventor : UMETANI YUKIO

(54) METHOD FOR SCHEDULING INSTRUCTION STRING

(57)Abstract:

PURPOSE: To provide a method for scheduling instruction strings usable in the RISC computers of a wide range in common.

CONSTITUTION: A DAG generation part 12 analyzes instruction dependence relation DAG (121) from an assembly program 2 and an SMD generation part 13 divides an original instruction string into plural strings respectively composed of a part of instructions to be successively executed and generates a string merging diagram(SMD) 131 for specifying the mutual dependence relation of the divided strings and the levels. An instruction scheduler 11 uses the DAG and SMD and searches the instruction string for which execution time is minimum or close to the minimum while rearranging the instruction strings corresponding to genetic algorithm.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-110768

(43) 公開日 平成7年(1995)4月25日

(51) Int.Cl.⁶

G 0 6 F 9/38
9/45

識別記号

3 1 0 F

庁内整理番号

9292-5B

F I

G 0 6 F 9/ 44

技術表示箇所

3 2 2 F

審査請求 未請求 請求項の数30 O L (全 20 頁)

(21) 出願番号 特願平5-255497

(22) 出願日 平成5年(1993)10月13日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 梅谷 征雄

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

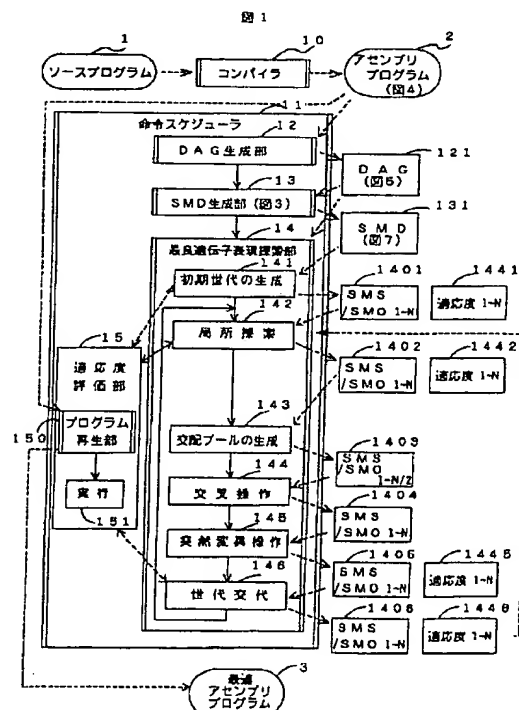
(74) 代理人 弁理士 小川 勝男

(54) 【発明の名称】 命令列スケジュール方法

(57) 【要約】 (修正有)

【目的】 広範囲のRISC計算機に共通して使える命令列のスケジュール方法を提供する。

【構成】 DAG生成部12がアセンブリプログラム2から命令依存関係DAG(121)を解析し、SMD生成部13が、このDAGから、元の命令列をそれぞれ順次実行されるべき一部の命令からなる複数のストリングに分割し、分割されたストリング相互の依存関係とそれらのレベルを指定するストリングマージ図(SMD)131を生成する。命令スケジューラ11はこのDAGとSMDとを使用して実行時間が最小あるいはそれに近い命令列を、遺伝的アルゴリズムに従って命令列を並べ替えながら探索する。



【特許請求の範囲】

【請求項 1】(a) プログラムを構成する命令列を解析して、それらの命令の間の依存関係を判別し、

(b) 遺伝的アルゴリズムに従って特定の計算機での該命令列の実行時間を少なくする、該判別された命令間の依存関係を満たす並び替え態様を探索し、その探索は、(b1) それぞれ該判別された命令間の依存関係を満たす命令並び替え態様を表す初期世代の複数の個体を生成し、(b2) 該複数の初期世代の個体を操作して、該特定の計算機での実行時間に関して、該複数の初期世代の個体のいずれか一つより改善された並び替え態様を表す複数の個体を含み、それぞれ該判別された命令間の依存関係を満たす命令並び替え態様を表す複数の次世代用の個体を生成し、(b3) 生成された複数の次世代用の個体を上記初期世代用の個体の代わりに使用してステップ(b2)(b3)を後続の複数の世代にわたって繰返し、(b4) 繰返し後に得られた複数の個体の内、該特定の計算機での実行時間に関して優れている一つの個体を表す並び替え態様を探索結果として選択するステップを有し、

(c) 該探索により得られた並び替え態様に従って該命令列を並び替えるステップを有する命令列スケジュール方法。

【請求項 2】ステップ(b2)は、

該複数の初期世代の個体で表される並び替え態様により該命令列を並び替えたときに得られる命令列の該特定の計算機での実行時間を評価し、

該複数の初期世代の個体に特定の操作を施し、新たな命令の並び替え態様を表す複数の変形個体を生成し、

該複数の変形個体で表される並び替え態様により該命令列を並び替えたときに得られる命令列の該特定の計算機での実行時間を評価し、

該複数の初期世代の個体と該複数の変形個体から、該特定の計算機での実行時間が相対的に少ない複数の個体を該次世代用の個体として選択するステップを有する請求項 1 記載の命令列スケジュール方法。

【請求項 3】該特定の操作を施すステップは、該生成された初期世代の複数の個体を組み合わせて複数の個体対を形成し、

それぞれの個体対に属する二つの個体を構成するデータの一部を交換して新たな一対の交叉変形個体を生成するステップを含む請求項 2 記載の命令列スケジュール方法。

【請求項 4】該特定の操作を施すステップは、該交換により生成された新たな複数の個体を構成するデータの一部をランダムに変更して、複数の突然変異個体を生成するステップをさらに含む請求項 3 記載の命令列スケジュール方法。

【請求項 5】該特定の操作を施すステップは、該複数の初期世代の個体の各々が表すデータの一部をその個体が生成された新たな複数の個体を構成するデータ

の一部をその個体が表す並び替え態様に近い並び替え態様を表すデータに変更することにより少なくとも一つの局所変形個体を生成し、

初期世代の各個体に対して生成された複数の局所変形個体で表される並び替え態様により該命令列を並び替えたときに得られる命令列の該特定の計算機での実行時間を評価し、

該複数の局所変形個体とその初期世代の個体に対して評価された実行時間に基づいて、該生成された複数の局所変形個体と初期世代の個体の一つを選択し、

上記生成、上記評価、上記選択を初期世代の他の複数の個体に対して繰り返し、もって複数の個体を選択し、選択された複数の個体を、該複数の初期世代の個体の代わりに該複数の個体対の生成に使用するステップをさらに有する請求項 4 記載の命令列スケジュール方法。

【請求項 6】該局所変形個体を生成するステップは、該複数の初期世代の個体の各々から複数の局所変形個体を生成するステップを有する請求項 5 記載の命令列スケジュール方法。

【請求項 7】該複数の初期世代の個体に関する該特定の計算機での実行時間を評価するステップおよび該特定の操作で生成された複数の個体に関する該特定の計算機での実行時間を評価するステップは、

評価すべき個体が表す命令並び替え態様にしたがって、該命令列を並び替え、

並び替えにより得られた命令列を該特定の計算機で実行して、その命令列の実行時間を計測するステップを有する請求項 2 記載の命令列スケジュール方法。

【請求項 8】初期世代の各個体は、各命令をマージすべきマージ先命令に関連する第 1 のデータと各命令のマージ順序に関連する第 2 のデータとを含む請求項 1 記載の命令列スケジュール方法。

【請求項 9】初期世代の各個体は、上記プログラムの上記命令列を構成する、それぞれ相互に依存関係がある複数の命令からなる複数の部分命令列(ストリング)に対応して設けられた、それぞれのストリング内の複数の命令の位置として、それぞれの命令をマージすべき他の命令を示す複数の部分からなる第 1 のデータと、該複数のストリングのマージ順序を示す第 2 のデータを含む請求項 1 記載の命令列スケジュール方法。

【請求項 10】初期世代の各個体の該第 1 のデータの各ストリングに対応する部分は、そのストリングに属する複数の命令のマージ先の命令として、他の一つ又は複数のストリングに属する複数の命令の位置を指定するデータからなる請求項 9 記載の命令列スケジュール方法。

【請求項 11】複数の初期世代の個体の生成ステップ(b1)は、

各初期世代の個体に対して該第 1 のデータと該第 2 のデータをランダムに決定するステップを有し、

その決定ステップは、

各初期世代の個体に対して、該複数のストリングの各々内の各命令をマージすべ位置として、いずれかの他のストリング内の命令を、該命令間の依存関係で許容される命令群の中からランダムに選択し、

該選択結果に基づいて、該第1のデータを決定し、該複数のストリングをマージすべきストリング順をランダムに決定し、

決定されたストリング順にしたがって、該第2のデータを決定するステップを有する請求項9記載の命令列スケジュール方法。

【請求項12】該初期世代に操作を施すステップは、該生成された初期世代の複数の個体を組み合わせて複数の個体対を形成し、

それぞれの個体対に属する二つの個体を構成するデータの一部を交換して新たな複数の交差変形個体を生成するステップを含み、

各個体対の間でのデータの一部を交換するステップは、その個体対の間で第1、第2のデータのいずれを交換するかをランダムに判定し、

その判定の結果、第1のデータを交換すると判定されたときには、該第1のデータの内、該複数のストリングの内のいずれに関する部分を交換するかをランダムに決定し、

その個体対の間でそれぞれの個体の第1のデータのうち、決定されたストリングに関する部分を交換し、上記判定の結果、第2のデータを交換すると判定されたときには、その個体対の間でそれぞれの個体の第2のデータを交換するステップを有する請求項11記載の命令列スケジュール方法。

【請求項13】該初期世代に操作を施すステップは、該データ交換により生成された複数の個体を構成するデータの一部をランダムに変更して、複数の突然変異個体を生成するステップをさらに含み、

該複数の突然変異個体を生成するステップは、該データ交換により得られた複数の交叉変形個体の各々について、その第1、第2のデータのいずれを変更するかランダムに判定し、

その判定の結果、その個体の第1のデータを変更すると判定されたときには、その個体の第1のデータのうち、該複数のストリングの内のいずれに関する部分を変更するかをランダムに決定し、

その個体の該第1のデータのうち、該決定されたストリング内のいずれの命令に関する部分を変更するかをランダムに決定し、

その第1のデータのうち、その決定された命令のマージ位置を表す部分を、該命令間の依存関係により許容される範囲内でランダムに変更し、

上記判定の結果、第2のデータを変更すると判定されたときには、その個体の第2のデータをランダムに変更するステップを有する請求項12記載の命令列スケジュー

ル方法。

【請求項14】該初期世代に操作を施すステップは、該複数の初期世代の個体の各々が表すデータの一部をその個体が生成された新たな複数の個体を構成するデータをその個体が表す並び替え態様に近い並び替え態様を表すデータに変更することにより少なくとも一つの局所変形個体を生成し、

初期世代の各個体に対して生成された複数の局所変形個体で表される並び替え態様により該命令列を並び替えたときに得られる命令列の該特定の計算機での実行時間を評価し、

該複数の局所変形個体とその初期世代の個体に対して評価された実行時間に基づいて、該複数の複数の局所変形個体と初期世代の個体の一つを選択し、

上記生成、評価、選択を初期世代の他の個体に対して繰り返し、もって、複数の個体を選択し、

選択された複数の個体を、該複数の初期世代の個体の代わりに該複数の個体対の生成に使用するステップをさらに有し、

各初期世代個体に対する該局所変形個体の生成は、該複数のストリングの少なくとも一つとそのストリング内の複数の命令の少なくとも一つを選択し、

その初期世代の個体の第1のデータの内、その選択された命令のマージ位置に関する部分を、その第1のデータが現に示すマージ位置の隣りの位置であって、上記命令間の依存関係により許容される位置を表すデータに変更した第1のデータを有する局所変形個体を生成するステップを有する請求項12記載の命令列スケジュール方法。

【請求項15】該局所変形個体の生成ステップは、各初期世代の個体に対して複数の局所変形個体を生成するステップを有し、そのステップは、

各初期世代の個体に対して、該複数のストリングがとり得るあらかじめ順序づけられた複数のストリング順のうち、その順から見て隣接する位置にあるストリング順を示すデータにその個体の第2のデータを変形した局所変形個体を生成するステップをさらに有する請求項14記載の命令列スケジュール方法。

【請求項16】判別された命令間の依存関係に基づいて、上記プログラムの上記命令列を、それぞれ相互に依存関係がある複数の命令からなる複数の部分命令列(ストリング)に、かつ、各命令がいずれか一つのストリングのみに含まれるように、分割し、

それぞれのストリングに含まれる複数の命令と他のストリングに含まれる複数の命令との間の上記判別された依存関係に基づいて、該複数のストリング間の依存関係を判別するステップをさらに有し、

初期世代の各個体の生成ステップは、該ストリング間の依存関係に基づいて、その個体を生成するステップを有する請求項1記載の命令列スケジュール方法。

【請求項 17】該複数のストリング間の依存関係に基づいて、各ストリングがストリング間依存関係の階層内に位置するレベルを各ストリングに対して決定するステップをさらに有し、

該ストリング間の依存関係に基づいて、初期世代の各個体を生成するステップは、

各ストリング内の複数の命令をマージすべきマージ先ストリングとして、そのストリングが依存関係を有し、そのストリングよりレベルの低い一つ又は複数の他のストリングを選択し、

各ストリングに対して選択された一つ又は複数のマージ先ストリングに基づいて、その個体を生成する請求項 16 記載の命令列スケジュール方法。

【請求項 18】各ストリングに対して選択された一つ又は複数のマージ先ストリングに基づいて、初期世代の各個体を生成するステップは、

各ストリング内の各命令のマージ位置として、そのストリングに対して上記選択された一つ又は複数のストリング内の、該命令間の依存関係によりマージが許容される範囲にあるいずれかの命令の位置を決定し、

該複数のストリングの複数の命令に対して決定されたマージ位置をあらわす第 1 のデータを含むデータをその個体をあらわすデータとして生成するステップを有する請求項 17 記載の命令列スケジュール方法。

【請求項 19】初期世代の各個体を生成するステップは、

該複数のストリングのマージ順序を示す第 2 のデータを生成し、

該生成された第 2 のデータを、その個体に対して決定された第 1 のデータとの組み合わせをその個体を表すデータと生成する請求項 18 記載の命令列スケジュール方法。

【請求項 20】初期世代の各個体を生成するステップは、

該複数のストリングをそれぞれのレベルの順に並べたストリング順を表すように第 2 のデータを生成するステップを更に有する請求項 19 記載の命令列スケジュール方法。

【請求項 21】(a) プログラムを構成する命令列を解析して、それらの命令の間の依存関係を判別し、

(b) 判別された依存関係に基づいて、この命令列を、相互に依存関係がある複数の命令を含む複数の部分命令列(ストリング)に、かつ、各命令がいずれか一つのストリングのみに含まれるように、分割し、

(c) 上記ストリングをマージするためのストリング順と、それぞれのストリングの複数の命令をそれぞれマージするための、それぞれのストリング以外の他のストリング内のマージ位置とからそれぞれなる複数の並び替え態様を順次指定し、

(d) 各並び替え態様により指定された、ストリング順と

該複数のストリングの複数の命令のマージ位置とに基づいて、該複数のストリングを一つの命令列にマージし、
(e) 各並び替え態様に対して得られた命令列を特定の計算機で実行するに必要な実行時間を評価し、

(f) 該複数の並び替え態様に対して得られた複数の命令列の一つを、該複数の命令列に対して、上記ステップ(e)により得られた評価結果に基づいて、上記特定の計算機で実行すべき命令列として選択する命令列スケジュール方法。

10 【請求項 22】上記各ストリングの複数の命令のマージ位置を指定するステップは、それぞれの命令をマージする位置として、そのストリング毎に定めた他の一つ又は複数のストリングのいずれか一つに属し、上記命令間依存関係を満たす位置を指定するステップを有する請求項 21 記載の命令列スケジュール方法。

【請求項 23】上記ストリングのマージするステップは、

上記指定されたストリング順の先頭に位置するストリングに、その先頭以外の他の複数のストリングを、この指定されたマージ順に順次マージし、

20 この際、該複数の後続のストリングの各々のマージにあたっては、そのストリング内の各命令に対して指定されたマージ位置に従って、その命令をそのストリングに先行する他のストリングのマージの結果得られるストリングに順次マージして一つの命令列を生成するステップを有する請求項 21 記載の命令列スケジュール方法。

【請求項 24】該複数のストリングの間の依存関係を、それぞれのストリングに含まれる複数の命令と他のストリングに含まれる複数の命令の間の上記判別された依存関係により判別し、

該複数のストリング間の依存関係に基づいて、各ストリングがストリング間依存関係の階層内に位置するレベルを各ストリングに対して決定するステップをさらに有し、

上記各ストリングの各命令のマージ位置を指定するステップおよびそれを変更するステップは、そのストリングがそれに対して依存関係を有し、上記決定されたレベルがより低い他の先ストリングに属し、上記判別された命令間の依存関係を満たすマージ位置を指定するステップを有する請求項 21 記載の命令列スケジュール方法。

【請求項 25】上記ストリング順のを指定するステップは、この判別されたストリングのレベル順に配列されたストリング順を選択するステップを有し、

上記ストリング順を変更するステップは、この判別されたレベル順に配列され、かつ、同一のレベルに属する複数のストリングの一部の順序が既に指定されたストリング順内の順序とは異なるストリング順を選択するステップを有する請求項 24 記載の命令列スケジュール方法。

【請求項 26】上記各ストリングのレベルを決定するステップは、最長のストリングを最下層のレベルのストリ

ングとして選択するステップを有する請求項 2 4 記載の命令列スケジューリング方法。

【請求項 2 7】上記複数の列び替え態様を順次指定するステップは、上記特定の計算機の構造に依存しない手順で該複数の列び替え態様を順次指定するステップを有する請求項 2 1 記載の命令列スケジューリング方法。

【請求項 2 8】上記特定の計算機の構造に依存しない手順により該複数の列び替え態様を順次指定するステップは、

一つのストリング順と各ストリングの各命令のマージ位置を表すデータとの組からなる複数の遺伝子的表現を生成し、

生成された複数の遺伝子的表現を、遺伝的アルゴリズムに依存する手順で変更するステップを有する請求項 2 1 記載の命令列スケジューリング方法。

【請求項 2 9】上記複数の遺伝子的表現を変更するステップは、

該生成された複数の遺伝子表現のデータの一部を相互に交換して複数の新たな遺伝子表現を生成する交叉操作により生成し、

交叉操作により生成されたいずれかの遺伝子表現の少なくとも一部を突然変異操作による変更するステップを有する請求項 2 8 記載の命令列スケジューリング方法。

【請求項 3 0】上記複数の遺伝子的表現を変更するステップは、複数の遺伝子表現のデータの一部を、それぞれそのデータが表わす命令の並び替え態様に近い他の並び替え態様を表わす他のデータに変更する局所探索による生成を含む請求項 2 9 記載の命令列スケジューリング方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、計算機の性能を引き出すための命令列の並び替え（スケジューリング）の方法に関し、特に RISC (Reduced Instruction Set Computer) に好適な、命令列スケジューリング方法に関する。

【0002】

【従来の技術】半導体製造技術と回路設計技術の近年の進歩を反映した RISC 計算機は複雑なパイプライン制御方式と命令起動方式を用いるため、命令列の実行時間はその並び順によって変化する。すなわち、命令列を許容範囲内で入れ替えてその性能を引き出す Sequence Optimizer が重要な役割を果たす。性能を引き出すためには、依存関係を有する命令同志が隣接して並ばないように引き離すことが基本的に重要であるのみならず、並列に動作可能な演算器の構成と命令の組合せに適合した最適な入替えを行う必要がある。例えば、Phillip B. Gibbons, etc., "Efficient Instruction Scheduling for a Pipelined Architecture", Proc. SIGPLAN '86 Symposium on Compiler Construction, pp. 11-16 (以下公知文献 [1] と呼ぶ) に代表されるように、従来の命令入替え方式では、依存命令の引き離しを目的とした幾つかの固定的

な規則のみを用いて並べ替えを行っていた。

【0003】

【発明が解決しようとする課題】上記の固定的な規則のみを用いた並べ替えではすべてのケースを尽くすわけではないので、十分な性能引出しがなされない場合がある。また計算機の論理構成や制御方式などの特性が変わる度に並び替え規則を変更する必要があった。本発明の目的は、スケジューリングすべき命令を実行する計算機の論理構成あるいは制御方式に依存しない並び替え手順で、命令の並び替えできる命令列スケジューリング方法を提供することである。

【0004】

【課題を解決するための手段】この目的達成のために、本願第 1 の発明による命令列スケジューリング方法は、(a) プログラムを構成する命令列を解析して、それらの命令の間の依存関係を判別し、(b) 遺伝的アルゴリズムに従って特定の計算機での該命令列の実行時間を少なくする、該判別された命令間の依存関係を満たす並び替え態様を探索し、(c) 該探索により得られた並び替え態様に従って該命令列を並び替えるステップを有する。

【0005】本願第 1 の発明のより具体的な態様は、この命令間の依存関係に基づいて、上記プログラムの上記命令列を、それぞれ相互に依存関係がある複数の命令からなる複数の部分命令列（ストリング）に、かつ、各命令がいずれか一つのストリングのみに含まれるように、分割し、それぞれのストリングに含まれる複数の命令と他のストリングに含まれる複数の命令との間の上記判別された依存関係に基づいて、該複数のストリング間の依存関係を判別し、該命令間の依存関係と該ストリング間の依存関係とに基づいて、遺伝子アルゴリズムで使用する初期世代の各個体を生成するステップを有する。

【0006】さらに、本願第 2 の発明による命令列スケジューリング方法は、(a) プログラムを構成する命令列を解析して、それらの命令の間の依存関係を判別し、(b) 判別された依存関係に基づいて、この命令列を、相互に依存関係がある複数の命令を含む複数の部分命令列（ストリング）に、かつ、各命令がいずれか一つのストリングのみに含まれるように、分割し、(c) 上記ストリングをマージするためのストリング順と、それぞれのストリングの複数の命令をそれぞれマージするための、それぞれのストリング以外の他の一つ又は複数のストリング内のマージ位置とからそれぞれなる複数の並び替え態様を順次指定し、(d) 各並び替え態様により指定された、ストリング順と該複数のストリングの複数の命令のマージ位置とに基づいて、該複数のストリングを一つの命令列にマージし、(e) 各並び替え態様に対して得られた命令列を特定の計算機で実行するに必要な実行時間を評価し、(f) 該複数の並び替え態様に対して得られた複数の命令列の一つを、該複数の命令列に対して、上記ステップ (e) により得られた評価結果に基づいて、上記特定の計

算機で実行すべき命令列として選択するステップを有する。

【0007】

【作用】本願第1の発明では、特定の計算機での実行時間を少なくするように、命令の並び替えの態様を、遺伝的アルゴリズムにより探索するので、探索の過程においては、その特定の計算機の論理方式や制御方式に立ち入る必要があるので、計算機特性の変化に幅広く対応して命令列をスケジュールできる。さらに、本願第2の発明では、命令列の並び替え状態を、ストリングのマージ順と、各ストリング内の命令のマージ位置により指定するので、この指定を順次切り替えることにより、計算機での実行時間に関して最適あるいはそれに近い、命令列の並び替え態様を決定出来る。

【0008】

【実施例】

(1) 実施例の概要

図2に本発明による命令列スケジュール方法を実施するための計算機システムの全体構成を示す。このシステムは、命令列のスケジュールを実行するための計算機システム1000と、スケジュールされる命令列を実行する、たとえばRISCタイプの計算機2000とからなる。前者は、中央処理装置(CPU)100、このCPUの主記憶装置として使用されるメモリ200、ファイル記憶装置300、キーボードなどの入力装置400、スケジュールの結果などを表示するためのCRT500とからなる。ファイル記憶装置300には、C言語等で書かれた、命令列のスケジュールを施すべきソースプログラム1を保持するとともに、それから後にコンパイルして得られるアセンブリプログラム2を保持するのに使用される。メモリ200には、ソースプログラム1をアセンブリするための、それ自体公知のコンパイラ10と、命令列のスケジュールをするためのプログラムである命令列スケジューラ11とが格納される。

【0009】本実施例では、このスケジューラ11が、アセンブリプログラム2を解析して、そのプログラムの構造を表す情報として、このプログラムに含まれる命令間の依存関係を表す命令依存解析図(Dependency analysis graph-DAG)121を得るとともに、このプログラムをそれぞれ複数の命令からなる複数のストリングに分割し、それぞれのストリングの相互の依存関係を表すストリングマージ図(String merge state diagram-SMD)131とを得る。これらの情報をベースにして、スケジューラ11は、このプログラムの命令列の並び替えを予め定めた一定の並び替え手順で行なう。具体的には、本実施例では遺伝子アルゴリズムに従って行なう。この際、それぞれの命令列の並びを表す情報として、ストリング内の命令のマージ位置を表す情報であるストリングマージ状態(String merge state-SMS)およびストリングのマージの順序を表す情報であるスト

リングマージオーダ(String merge order-SMO)との組み合わせ1401、、1407を使用するようになっている。各並び替えの結果については、プログラム再生部150によりその並びに対応したアセンブリプログラムを生成し、実計算機2000(図2)で実際に実行して、その適応度としての実行時間を測定し、その並び替えの良否の評価に使用し、その後の並び替えに反映し、最終的には最適な並び替えを決定するようになっている。

【0010】とくに、本実施例では、命令列の並び替え態様を表すデータを遺伝的アルゴリズムに従って、操作して、最適又はほぼ最適な命令列並び替え態様を決める所に特徴がある。より具体的には、プログラムを命令間の依存関係に基づいてストリングに分けて、各ストリング間の依存関係に基づいて各ストリングのマージ先ストリングを決め、命令間の依存関係の制約の下で許される範囲で、ストリングの各命令のマージ位置を変え、ストリング間の依存関係の制約下で許される範囲でストリングのマージ順序を変えることにより、命令列の並び替えを種々行ない、得られる種々の命令列の良し悪しは、その計算機の特性を考慮して判別する。これらの操作が遺伝的アルゴリズムに従って行なわれる。

【0011】その他の特徴は以下で説明する。

【0012】(2) 遺伝的アルゴリズムによる命令列並び替え処理の概要

遺伝的アルゴリズム(Genetic Algorithm)それ自体は、公知である。例えば、D. E. Goldberg: "Genetic Algorithm in Search, Optimization and Machine Learning", Addison-Wesley, 1989, pp.1-23(以下参考文献[2]と呼ぶ)ないし H. Muehlenbein, D. Schlierkamp-Vossen: "Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization", Evolutionary Computation 1, MIT Press, 1993, pp.1-18(以下参考文献[3]と呼ぶ) 参照。遺伝的アルゴリズムは、生命体の進化・適応過程の数学的モデルであり、適当な遺伝子で表現される個体集団に対して、局所探索(後天的適応)、交配(交叉)、突然変異などの遺伝的操作を施し、その結果から新世代の個体を生成し、それに対して以上の処理を繰返し実行し、こうして、世代を追って集団としての改良を図って行くものである。従って、遺伝的アルゴリズムを命令の並び替えに適用すれば、原理的に命令列の許される全ての並べ替えを候補とすることができ、そこから効率良く最良のものを探索するので、従来方式に較べて良いものが選ばれる可能性が高い。また探索の過程で用いる適応度の評価尺度に計算機特性を反映しておけば自動的に尺度に応じた最良解が選択されるので、計算機特性の変化にも幅広く対応できる。

【0013】遺伝的アルゴリズムを命令列の並べ替えに適用する上で考慮すべき点は、遺伝子として使用する"命令の並べ替え状態"の表現形式と遺伝的操作の定義で

あり、これらが探索の質と効率を大きく左右する。命令の並べ替え状態を表す素朴な表現は命令番号を出現順に並べた数値列(順列表現)である。これに通常の遺伝的操作である交叉や突然変異を行うと、これらの操作は、元の数値を部分的に入れ替えたり他の値に変更するだけであるので、これらの操作の結果得られる数値列は、複数の位置に同じ値をもつ数字列となり、もはや順列表現ではなくなる。従ってこの結果に対するその後の補正が必要である。本実施例ではこの点を考慮して命令の並びを表すデータとして遺伝的アルゴリズムに適したもの(以下、これを簡単化のために遺伝子表現という)を工夫した。

【0014】なお、遺伝的アルゴリズムの応用にあたっては、一般的には、改良の程度を示す尺度となる適応度関数の決定も問題となるが、命令列の並び替えへの適用にあつては、適応度関数としては、並べ替えられた命令列を特定の計算機で実行したときの実行時間ないし実行サイクル数を使用する。従来から命令列を構成する命令間の依存関係を表現するために命令解析グラフ(Dependency Analysis Graph-DAG)と呼ぶグラフが用いられ、命令列の並べ替えもDAGの表現する依存関係の制限内で行われなければならない。本実施例でもこのDAGは使用する。本実施例では、新たな概念としてストリングと呼ぶ命令間の依存関係でつながれた、元の命令の一部の命令からなる部分列を導入する。ストリングはDAG上の一次元の部分グラフであり、元の命令列は、DAGに依存して、複数のストリングに分解できる。各命令はいずれか一つのストリングに属し、複数のストリングに重複して属さない。この意味で、各ストリングは他のストリングに対して排他的である。ストリングは依存関係でつながれた糸状の命令の列であるので、それを続けて実行すると、その内の一つの命令が終わらないと次の命令を実行できない。この結果、RISC計算機内のパイプラインに空きを生じ、実行性能が低下する。そこで複数のストリングをマージして実行することがスケジューリング上必要である。

【0015】そのためストリング間のマージ状態を表現するコードとしてストリングマージ状態コード(String Merge Status-SMS)を導入する。SMSコードの内容はストリング間のマージ構造(どのストリングをどのストリングにマージさせるか)に依存するが、できるかぎりマージの階層数を減らし、ストリングごとのコードの設定が独立にできるようにした方が表現の冗長性が少なく、遺伝的アルゴリズムによる探索の効率をあげることが期待できる。そのためDAGで表現された命令間の依存関係で結ばれたストリングの間でのみマージ関係を作ることとした。具体的には、各ストリングのSMSコードは、そのストリングの各命令をマージする、マージ先ストリング内の位置を表す。全ストリングのマージ構造を決めるために、すべてのストリングがそこにマージ

される基底のストリングを一つ定めた後、そこからの逆マージ系列の最長の長さによりストリングにレベルを付け、ストリング全体を階層的に組織化する。このストリング間の階層構造をストリングマージ図(String Merge Diagram-SMD)と名付ける。

【0016】さらにこの階層構造の中で、複数のストリングが共通のマージ先ストリングを持ちかつマージ点の競合がある場合に優先度を定めるため、ストリングの優先度リストであるストリングマージ順序(String Merge Order-SMO)とよぶ別のコードを導入する。SMOコードは、ストリングの名称をレベル順に並べたものである。DAGの制約内での命令列の並び替えは各ストリングを構成する命令の順番を変えずストリング同志のマージ(混合)状態のみを変えるので、命令列の並べ替え状態は、ストリングごとのSMSコードとひとつのSMOコードで表現できる。これを遺伝子表現に用いる。

【0017】遺伝的アルゴリズムを実現するためにはさらに、局所探索(Local Search)、交叉(CrossoverまたはRecombination)、突然変異(Mutation)等の遺伝的操作(Genetic Operation)を定義する必要がある。本実施例では上記の遺伝子表現を用いて、交叉操作を、2つの遺伝子表現の間での特定ストリングのSMSコードの交換ないし特定レベル部分のSMOコードの交換と定義する。同じく突然変異操作を、選択したストリング群のSMSコードないし選択したレベル群のSMOコードの現在値からの変更と定義する。局所探索操作はSMSコードないしSMOコードのうちの1ヶ所の、適応度の改善をもたらすような隣接値への変更と定義する。各ストリングの各命令のマージ位置を示すSMSコードと、ストリングのマージの優先度を示すSMOコードとの性質から、上記の遺伝的操作の結果は、自動的に後補正の必要がない。さらに、遺伝的アルゴリズムに従い命令列の並び替えを行なうので、特定の計算機の構造に依存しない手順で命令列の並び替えを行なうことが出来る。遺伝的アルゴリズムを実現するためにはまた、適応度の評価が必要であるが、本実施例では、並べ替えられた命令列の特定RISC計算機での実行時間で図った実行サイクル数を適応度とする。当然、その値が小さいほど高い適応度を意味する。

【0018】(3)スケジューラ11の詳細

図1に従い、本実施例の処理の詳細を説明する。ソースプログラム1からコンパイラ11により得られたアセンブリ語の命令列(アセンブリプログラム)2が得られる。図4にソースプログラム1をコンパイルして得られるアセンブリプログラム2の1例を示す。この表記では左から、命令番号、ラベル、命令コード、第1入力データを保持する特定のRISC計算機内のレジスタの番号、第2入力データないしそれを保持するレジスタ番号、出力データを保持するレジスタ番号を示す。6, 8, 11, 18番命令のような主記憶装置からのロード/ストア命

令では、鍵カッコのなかはアドレスレジスタの番号を示す。

【0019】アセンブリプログラム2は、命令スケジューラ11により、以下のように処理される。

(3-1) DAG生成部12

これにより、このアセンブリプログラム2に対するDAG121が生成される。DAG121の生成は、たとえば、Jeanne Ferrante "The Program Dependence Graph and Its Use in Optimization", ACM Transaction on Programming Language and Systems, Vol.9, No.3, July 1987, pp.319—349(以下参考文献[4]と呼ぶ)などに示される公知の方法で行われる。図4のアセンブリプログラムから生成されたDAG121を図5に示す。ここで丸括弧内の数字は命令番号を示す。線(アーク)は命令間の依存関係を示し、線の上の命令の結果を下の命令が使うか上下の少なくとも一方が分岐命令であること、従って上の命令が下の命令に先立って行われるべきことを意味する。DAG121の生成は通常のコンパイラで実施しているとおり、命令間のレジスタの参照関係や演算命令と分岐命令等の制御命令の順序関係を解析して行われ

【0020】(3-2) SMD生成部13

続いてSMD生成部13がDAG121よりSMD(String Merge Diagram: スtringマージ図)131を生成する。その手順を図3を参照して説明する。

(ステップ31) まず、DAG121をStringに排他的に分解し、その結果としてStringベクトル(String Vector-STV)311(図6)を得る。StringはDAGで示される命令間の依存関係のアークで順次つながれた複数の命令からなる部分命令列である。命令列のStringへの分解は以下の手順で行われる。

【0021】まずDAG内で上からの依存関係を持たない任意の一つの命令(例えば、命令(1))を選択し、そこを始点としてアークの一つを任意に選択しながら下方にアークが尽きるまで辿って第1のString(この場合は(A))を抽出する。続いて未抽出の命令からの依存関係を持たない命令を任意に一つ選び、そこを始点として未抽出命令への下方アークが無くなるまでアークの一つを任意に選んで辿る操作を繰り返して、残りのString(この場合は(B),(C),(D),(E))を抽出する。例えば(B)については始点命令(7)からアークをたどって(8)を抽出するが次の(9)は既抽出であるのでここで終わりとなる。Stringの抽出には任意性があるがそれで構わない。図5のDAGに対して得られたSTVの例を図6に示す。STVは2次元のリスト構造で保持する。図6でアルファベットはStringの名称、続く番号列はStringを構成する命令列を示す。

【0022】(ステップ32) 図3に戻り、このステップで、Stringのレベルを決定する。各Stringのレベルは、String間のマージ状態を階層的に整理し

て管理するために導入するものであり、低レベルのStringに高レベルのStringがマージされるように、マージ時に使用する。レベル付けは命令間の依存関係から派生するString同志の依存関係に基づいておこなう。各々のStringに属する命令の間にDAGで規定される依存関係があるとき、String同志は依存関係を持つという。たとえば図6でString(A)の命令(9)はString(B)の命令(8)と図5のDAG上依存関係があるので(A)と(B)は依存関係を持ち、(B)と(C)は持たない。各Stringのレベルは、より低位のStringのレベル値がより小さくなるように決める。

【0023】具体的には、まず、最低位のStringを選びそれにレベル1を与える。レベル付けは、最低位のStringに何を選ぶかにより任意性があるが基本的に何であっても構わない。ここでは最も長いString(A)を選びそれにレベル1を与える。その理由は以下のとおりである。後に説明するように、各Stringにマージコード(SMSコード)を付与する。その際、最低位のStringにマージコードが不要である。さらに、各StringのSMSは、そのStringの命令の数に応じて長くなる。処理の高速化のために、StringのSMSコードの最大長をを短くすることが望ましい。従って、最も長いString(A)を選びそれにレベル1を与えることは、処理の高速化の上で望ましい。

【0024】続いて最低位のレベルのString(A)と依存関係を持たないStringがあればそれにもレベル1を与える。図6の例では、そのようなStringはない。続いてレベル1のStringあるいはレベルの与えられていないStringとは依存関係はあるが、それらのString以外のStringとは依存関係のないString String、例えば、(B),(C),(D)にレベル2を与える。さらにレベル1, 2と依存関係を有するが、それ以外のStringとは依存関係を有しないString(E)にレベル3を与える。以下同様にこの手順をすべてのStringにレベルが与えられるまで続ける。

【0025】(ステップ33) 最後にステップ33により、Stringのレベル情報とDAG121にもとづいて、Stringを階層化して図7に例示するような情報を有するStringマージ図(SMD)131を生成する。このSMD131は、少なくとも、各Stringを構成する命令とそれらの実行順と、各StringのレベルとString間の依存関係を示す情報を有する。図7の矢印は依存関係を有するStringの間で高レベルから低レベルに向かって付けられ、2つのStringがマージ元Stringとマージ先Stringの関係にあることを明らかにする。このStringの依存関係は、以上のレベルの決め方から明かであるように、DAG121による命令の依存関係に依存して決められている。なお、図6の例では、String(B)(C)(D)はString(A)にのみ依存するが、String(E)は、String

グ(B)に依存するだけでなく、ストリング(A)にも依存する。こうしてSMD生成部13の処理が終了する。

【0026】(3-3) 命令列の並び替え状態を表すデータ

図1に戻り、最良遺伝子表現探索部14は、生成されたDAG121とSMD131を用いて、命令列の並びを表すデータ(遺伝子表現)を生成し、さらに、遺伝的アルゴリズムに従ってその遺伝子表現を順次更新し、それにより命令列最良遺伝子表現の探索を行う。最良遺伝子表現探索部14の詳細説明の前に、本実施例で使用する命令列の並びを表す遺伝子表現を先に説明する。

【0027】本実施例では、命令の並び替えは、プログラムを構成する複数のストリングの各々の命令を他のストリングにマージする位置を変えて、かつ、ストリングのマージ順をかえて行なう。このような命令の並び替えを行なうために、本実施例では、全てのストリングに対するストリングマージ順序コード(SMOコード)と、全ストリングに対する一群のストリングマージ状態コード(SMSコード)との組を使用し、かつ、この組を遺伝的アルゴリズムにより命令列を並び替えるために、命令列の並びかえ状態を表すデータ(遺伝子表現)として使用する。

【0028】遺伝子表現の例を図8に示す。図において、SMSコード群81は、プログラムを構成する複数のストリングのSMSコードの集合である。各ストリングは一つのSMSコードを有し、そのSMSコードは、そのストリング内の複数の命令をそれぞれマージすべき、他ストリング内の複数の位置を示すデータからなる。SMOコードはプログラムを構成する複数のストリングをマージする順番を指定する。各ストリングのSMSコードは、そのストリングを構成する複数の命令の各々が挿入されるべき、マージ先ストリング内の位置を表すデータからなる。本実施例では同一ストリング内の複数の命令は、他の一つのストリングにマージするように、それらのSMSコードを決めるこれらの位置は、そのマージ先ストリング内のマージ位置の直前に来る命令の番号で表現する。なお、マージ先ストリングの先頭に挿入する場合は命令番号としてコード0を使う。

【0029】例えば、図8に示した例では、ストリング(B)のSMSコード(6-6)は、第一命令(7)、第二命令(8)共にマージ先ストリング(A)の命令(6)の後に挿入される事を意味する。同様に、ストリング(C)のSMS(9-12)は、第1命令(10)、第2命令(12)がそれぞれストリング(A)の命令(9)、(12)の後に挿入されることを示し、ストリング(D)のSMS(12-12)は、その第1命令(14)、第2命令(16)が共にストリング(A)の命令(12)の後に挿入されることを意味する。これらのストリング(B)(C)(D)は、いずれもストリング(A)にしか依存関係を有しない。

【0030】後に説明するように、本実施例では、ある

マージすべきストリングが依存関係を有する他のストリングをマージ先ストリングとして使用する。マージすべきストリングが依存関係を有する他のストリングが複数あるとき、それらをマージ先ストリングとして使用する。より具体的には、本実施例では、例えば、図8に示した例では、ストリング(E)は、図7に関して説明したように、ストリング(A)(B)に依存する。このストリング(E)のSMS(12)は、ストリング(E)内の命令(13)を、実際のマージにあたっては、ストリング(A)に(B)をマージした後に得られるストリング中の命令(12)の後に挿入される。図7に関して述べたように、ストリング間の依存関係および各ストリングのレベルはSMDから判別可能である。従って、各ストリングのSMSは、このSMDから決定できる。

【0031】SMOコードは、ストリング名の順列からなる。プログラムを構成する複数のストリングをマージして命令列を構成するには、ストリング間の依存関係を満たすようにマージするストリングの順番に制限を設けなければならない。このSMOコードは、このようなストリングをのマージする順番を指定する。ストリングのレベルの決め方から分かるように、ある第1のストリングに依存関係のある他の第2のストリングのレベルは、その第1のストリングのレベルより高くなるように決めた。したがって、本実施例では、複数のストリングのマージは、それらのストリングの内、レベルが最低のものに、他のストリングをレベルの低いものから順にマージするように行う。したがって、SMO内では、低レベルストリングが高レベルストリングに先行しなければならない。図8のSMOコード82では、(A)がレベル1、(B)、(C)、(D)がレベル2、(E)がレベル3なので、SMOコード82は、この制約を守っていることが分かる。

【0032】SMOを決定するに必要な情報はSMDから得られることもSMSの場合と同じである。

【0033】なお、ストリングの切りだし方およびレベル1のストリングの決定には、任意性があることは既に述べたとおりである。本実施例では、あるストリングを他のストリングにマージするときに、当該他のストリング内の先頭の命令より前に、そのマージすべきストリング内の命令もマージすることも可能になっている。勿論、この位置にマージすることがDAGにより許される場合に限る。これにより、マージされるストリング内の命令より前の位置に、マージすべきストリング内の命令をマージ可能になっている。従って、マージするストリングの順序を決めても、マージされる結果は、このマージされる順序に制約されないようになっている。この結果、ストリングのマージ順序を制限しても、その順序に制限されない位置に命令が位置することが可能になっている。

【0034】以上から分かるように、このように定めた

SMS群とSMOとの組み合わせは、異なる命令の並びに対しては、異なる値を有することが分かる。したがって、特定のSMS群と特定のSMOとの組み合わせが一つの命令の並びを表し、さらに、これらの値を替えることによりいろいろの命令の並び替え状態を表すことが出来る。

【0035】(3-4) SMS群とSMOとの組からの命令列の再生

本実施例では、実行時間の短い命令列（最適あるいはほぼ最適命令列）に対応するSMS群とSMOを求めた後、これらから、その命令列を再生する方法を取る。この再生処理は、この最適命令列の再生以外にも、本実施例ではこの最適命令列を遺伝アルゴリズムに従って探索する過程でも使用される。すなわち、この探索の過程で候補に上がったSMS群とSMOとの特定の組を評価するために、その組に対応する命令列を再生し、それらを実際に実計算機で実行させ、その実行時間を測定し、この候補の良し悪しを評価するようになっている。

【0036】この再生処理手順の理解は、SMS群とSMOとの組について上に述べた意義を理解するのに役立つので、ここで再生処理の手順をここで説明する。この再生処理は、図1の適応度評価部15内のプログラム再生部150によりおこなわれる。まず、SMOで指定される順番にストリングを選択する。すなわち、最初にレベルの最も小さいストリング、実施例ではストリング(A)を選択する。その後、次のレベルのストリングを選択し、後で選択されたストリングをSMSの指示に従って既に選択されたストリングにマージしてゆけば良い。

【0037】図8に例として示す、SMO82とSMS群81とから命令列を再生するには、次のようにすればよい。まずSMOから先頭のストリング(A)を構成する命令列を取り出す。つぎに、SMDの次のストリング(D)を取り出し、それを構成する命令(14)と(16)との組を、このストリングのSMS(12-12)に従って、ともにストリング(A)の命令(12)の後にマージする。実際には、まず、命令(14)を命令(12)の後にマージし、その後に命令(16)をマージする。すなわち、同じストリングに属する命令(14)と(16)の順序は入れ換えない。こうして新たな第1のストリング(1)-(2)-(3)-(4)-(5)-(6)-(9)-(11)-(12)-(14)-(16)-(18)-(17)を得る。

【0038】について、SMOの次のSTRING(C)を取り出し、そのSTRINGを構成する命令(10)と(15)を、SMS(9-12)に従って、この得られた第1の新たなSTRINGのそれぞれ命令(9)、(12)の後にマージする。こうして第2の新たなSTRING(1)-(2)-...-(9)-(10)-(11)-(12)-*

* (15) — (14) — (16) — (18) — (17) をえる。同様にして SMO の次のストリング (B) を取り出し、それを構成する命令 (7)、(8) を、SMS (6-6) に従って、この第 2 の新たなストリングの命令 (6) の後にマージして (1) — (2) — ... — (6) — (7) — (8) — (9) — (10) — ... — (16) — (18) — (17) をえる。最後にストリング (E) を構成する命令 (13) を、この SMS (12) に従って、第 3 の新たなストリングの命令 (12) の後にマージして最終的なストリング (1) — (2) — (3) — (4) — (5) — (6) — (7) — (8) — (9) — (10) — (11) — (12) — (13) — (15) — (14) — (16) — (18) — (17) を得る。こうして SMS 群と SMO とから、命令列の並びが、命令番号の列として決定された。この決定された並びにしたがって、プログラム再生部 150 は、それぞれの命令番号を有する命令をもとのアセンブリプログラム 2 (図 1) より取り出すことにより実際に並び替えた命令列からなるプログラムを再生する。

【0039】(3-5) SMS群とSMOの取り得る値
SMOで規定されるストリングのマージ順序のうち、同一のレベルのストリングのマージ順序は変更しても、レベルの低いストリングにレベルの高いストリングをマージするべきと言う上記制約には反しない。従って、同じレベルのストリング、この実施例では、ストリング(B)(C)(D)、の順序を入れ換えて複数のSMOを作ることが可能である。具体的には、本実施例では、以下のような6つのSMOを使用することが出来る。

【0040】各ストリングの命令の他のストリングへのマージ位置は、DAGで規定される命令間の依存関係により定まる許容範囲内で変えることができ、それに応じて異なるマージ状態すなわち命令並び替え状態を実現できる。従って、各ストリングのSMSコードも、これらの異なる並び替えに対応していろいろの値を持ち得る。例えば、ストリング(B)の命令(7)(8)については、DAG上命令(7)が(8)より先に実行されねばならず、命令(8)が(9)より先に実行されねばならないという制約がある。従って、ストリング(B)の命令(7)、(8)は、ストリング(A)上の先頭から命令(6)の後までの範囲にマージ可能である。従って、ストリング(B)の、ストリング(A)に対するSMSコードは、0-0から6-6の範囲(ただし後者の番号は前者の番号以上)で変更可能である。

【0041】同様に、ストリング(E)の命令(13)のストリングに対するSMSコードは、(B)を(A)にマージした結果のストリング上の命令(7)から(18)の範囲で変えられる。

【0042】

【表 1】

SMO#1	(A)(B)(C)(D)(E)
SMO#2	(A)(B)(D)(C)(E)
SMO#3	(A)(C)(B)(D)(E)
SMO#4	(A)(C)(D)(B)(E)
SMO#5	(A)(D)(B)(C)(E)
SMO#6	(A)(D)(C)(B)(E)

ここで、SMOの番号はこれらの異なる値を識別するために便宜的に表示するものであって、SMO自体はあくまでストリングの名称の列からなる。

【0043】(3-6) 最良遺伝子表現探索部14

図1に戻り、最良遺伝子表現探索部14の処理を以下に述べる。ここでは遺伝的アルゴリズムとして、ブリーダーアルゴリズム(Breeder Algorithm)(参考文献[3])と呼ぶアルゴリズムを用いている。この遺伝的アルゴリズムは、生命体の進化・適応過程の数学的モデルであり、適当な遺伝子で表現される個体集団の各世代に対して、交配(交叉)、突然変異、局所探索(後天的適応)などの遺伝的操作を施すことにより、世代を追って集団としての改良を図って行くものである。本実施例は、このアルゴリズムを命令のスケジュールに使用する。従って、以下でも個体、世代などのような遺伝的アルゴリズムで使用される用語を用いる。以下では、個体は命令列の特定の並びを表し、その個体を特定するデータ(遺伝子表現)は、既に述べたSMS群とSMOとの組で表す。

【0044】(3-6-1) 初期世代の生成処理141
まず始めにDAG121とSMD131を用いて初期世代のN個の個体(今の例では命令列の並び)の遺伝子表現1401(これはSMS群とSMOとからなる)の生成を行う。併せて、それぞれの個体の適応度1441を計測する。各個体の各ストリングのSMSコードは、SMD(図7)で決まるマージ先とDAG121で規定される命令のマージ位置の許容範囲に基づいて乱数を用いて決める。具体的には各個体の各ストリングのSMSコードの発生にあたっては、プログラムを構成するストリングをそれらのレベル順に上記したSMD(図7)から取り出す。先頭のレベル1のストリング、今の例では(A)、を取り出したときには、このストリングはSMSを要しないので、以下の処理をしないで、つぎのストリングを上記SMDから取り出す。取り出されたストリングを構成する複数の命令をそのストリング内の並び順にSMDから取り出す。取り出された命令の、そのストリングのマージ先ストリング(これは図7に示されるように、SMDにより定まる)内の位置を、DAG121で許容される範囲内から乱数により決定する。この処理をストリング内の命令が尽きるまで繰り返す。

【0045】たとえば図7のSMDで示されるストリングの場合、レベル1のストリング(A)はSMSコードが不要なのでスキップし、まずレベル2の(B)に取り組む。このストリングのマージ先ストリングは、図7に示すSMDからストリング(A)である。ストリング(B)の

第1命令(7)のマージ位置の許容範囲は、図5のDAG121から、マージ先ストリング(A)上の命令(1)から(6)の間なので、その範囲から一つのマージ位置の命令、たとえば命令(2)をランダムに選択する。第2命令(8)のマージ位置の許容範囲はストリング(A)上の、命令(7)用に選択した命令(2)から(6)までの間であることがDAG121から判別される。従って、この命令(8)のマージ位置として、その許容範囲から一つのマージ位置、たとえば命令(5)をランダムに選択する。この結果、この個体に関する命令(B)のSMSコードが(2-5)と決定された。続いてストリング(C),(D)のSMSコードを同様に決定する。最後に(E)の命令(13)のSMSコードを(A)に(B)をマージしたストリング(1)-(2)-(7)-(3)-(4)-(5)-(8)-(6)-...-(18)-(17)上の命令(7)から(18)の間でランダムに選択する。

【0046】各個体のSMOコードは、例えば、図8に示した例では、スケジュールすべきプログラムを構成するストリング(A)(B)(C)(D)(E)の内、同一レベルに属するストリング(今の例では(B)(C)(D))の範囲内でこれらのストリング(今の例では(B)(C)(D))の並びを乱数を用いて決める。すなわち、先に説明したSMOがとり得る6つのSMOのいずれかを乱数で選ぶことになる。以上の結果、一つの個体のSMS群とSMOが決定された。このような試行をあらかじめ規定された個体数であるN回分繰り返してN個の個体に対する遺伝子表現1401を生成する。

【0047】その後、各個体について適応度を計測するために、適応度評価部15を起動する。適応度として、各遺伝子表現が表す命令列の並びを有するプログラムの実行時間を使用する。勿論実行時間が少ない遺伝子表現の方が適応度が高い。この実行時間のため、適応度評価部15内のプログラム再生部150が、各個体の遺伝子表現から、それが表す命令列を有するプログラムを再生する。その手順は既に述べたとおりである。さらに、実行部151が、この再生された命令列は実計算機2000(図2)により実際に実行させ、実行完了に要した時間を計測する。各個体に関する実行時間の計測結果は、適応度データ1441として、後の使用のために記憶する。

【0048】続いて世代を追って個体を改良してゆく。

【0049】(3-6-2) 局所探索処理142

始めに上で得られたN個の個体に対して得られたN個の遺伝子表現1401に対して、局所探索による改良を行

い、改良後の新たな遺伝子表現1402を生成する。局所探索142では、N個の初期世代の個体のそれぞれの遺伝子表現を構成するSMSコード群ないしSMOコードのいずれか一つのコード値を隣接値に変更し、変更前の個体の適応度より高い適応度を有する個体が見つければ、その変更後の個体でもって、変更前の個体を置き換える。上記隣接値への変更を、その個体のいろいろのストリング、SMO内のいろいろのレベルの部分に対して繰り返して行く。ここで、あるストリングのSMSコードの隣接値とは、そのストリング内の特定の命令をマージする、マージ先ストリング上の、そのSMSが指定している位置を、隣接する位置に変更したときのSMSである。この隣接位置は、そのマージ先ストリング上の上流側の隣接命令（もとのSMSが指定するマージ位置にある命令より先に実行すべき命令）もしくは下流側の隣接命令のいずれかの位置である。但し、この隣接値への変更は、その隣接値にその特定の命令をマージすることが命令依存関係により許されている場合にのみ行なう。図9(a)は、局所探索操作により、SMSコードを隣接値に変更した例で、ここではストリング(C)の第一命令である(10)のSMSコード9を、ストリング(C)のマージ先ストリング(A)内の、そのSMSが指定する命令(9)の上流側で隣接する命令(6)に対する隣接値6に変更している。

【0050】SMOの隣接値とは、SMOが示すストリングの並びを、辞書の見出し順に並べたときの隣接するSMOのことである。隣接位置は、見出しの上流側で隣接する位置（見出しに先に表れる位置）あるいは下流側で隣接する位置のいずれかである。本実施例で使用している命令列の例では、SMOは6つの値を取ることが出来、それらは先に、表1にSMO#1からSMO#6として略記して示した。そこで示した6つのSMOの値の並びは辞書の見出し順になっている。図9(b)は、SMOを隣接値に変更した例を示す。ここで、レベル2のストリング(C)と(B)の順番を入れ替え、左側のSMO（これは表1のSMO#1に相当する）をそれに隣接するSMO#2に変更している。

【0051】以下具体的に、この局所探索の処理を説明する。なお、局所探索で変更すべきSMSあるいはSMO内の部分を決定する手順は、実質的に初期個体の生成の場合の同じである。まず各個体について、そのストリングをレベルの順に一つずつ選択し、選択したストリングの各々についてSMSの変更に関する以下の処理を行なう。但し、レベル1のストリングはSMSを有しないので、以下の処理は不用であるからである。次に、選択された一つのストリングを構成する複数の命令を順に一つずつ選択する。これらの命令を選ぶ順は、それらの命令のそのストリング内での順に従う。

【0052】こうして現在選択されている一つのストリングと、そのストリングから現在選択されている一つの

命令に関するSMSコードを、現在選択されている命令の現在のマージ位置の上流側の隣接するマージ位置に対するSMSへの変更を試みる。すなわち、その隣接位置への変更がDAGにより許されるかを判断し、許されるならば、この変更を行ない、許されないときにはこの変更は行なわない。こうして、選択されたストリングの変更後のSMSと他のストリングの未変更のSMSとからなるSMS群と、もとのSMOとの組からなる新たな個体を得る。この新たな遺伝子表現の適応度の評価を、先に初期世代の個体の場合と同様に行なう。上のSMSの変更と同じことを、上記現在選択されている命令の現在のSMSが指定するマージ位置の下流側で隣接するマージ位置に対するSMSへの変更を同じように行なう。

【0053】さらに、以上を、この選択されたストリングの他の命令に関して順次繰り返す。この繰返しときには、先に実行されたSMSの変更は、無視し、新たな変更は、操作を受けている個体の最初のSMS群に対して行なう。このことはこの局所操作に関する以下の処理でも同じである。さらに、以上と同じことをストリングを変えて行なう。こうして、レベル2以上の全てのストリングの全ての命令に対して以上のことが行なうことによりSMSの変更は終了する。

【0054】例えば、図8のストリング群の例では、レベル1のストリング(A)はSMSをもたないのでスキップし、まずレベル2のストリングの一つ(B)の最初の命令7に対するSMSコードである6を、現在のマージ先命令6の上流側の隣接の命令5をマージ位置とするSMSへ変更する。この変更は、DAGにより許され得る変更である。続いて、同じストリングの同じSMSを、命令6の下流側の命令9をマージ位置とするSMSに変更することを試みるが、命令7は、DAGにより、命令9より先に実行されねばならないから、このマージ位置の変更は行なわない。ついで、ストリング(B)の次の命令8に関するコード6の変更を試みる。ついでストリング(C),(D),(E)の順に進む。なお、ストリング(C)の最初のSMSコード9を6に変更する処理の結果は、図9(a)に示したとおりである。

【0055】この後は、SMOを隣接値に変更する。SMOの隣接値への変更は、SMOをレベルの低いグループの順に行なう。例えば、レベル1のストリングが複数ある場合には、他のレベルのストリングののマージ順は変更しないで、レベル1のそれらの複数のストリングのマージ順を、辞書見出しの上流側に隣接するSMOに対するものに変更する。現在処理中の個体の元のSMS群と、この変更後のSMOとの組により表される新たな個体を得る。この新たな個体について適応度を、前述の方法により評価する。以上を辞書の見出しの下流側で隣接するSMOに対するものに変更する。その後、レベル2以降のレベルの各々について同様の処理をする。もし、

レベル1のストリングが一つしかない場合には、レベル1に関して上に説明した処理を行なうことなく、レベル2以降のストリングの処理に移る。

【0056】例えば、図8のストリングの例では、SMOの第1レベルグループのストリング(A)、第2レベルグループのストリング列(D)-(C)-(B)、第3レベルグループの(E)の順にそれぞれのグループ内のストリングの入れ替えを試みるが、この例では、ストリングの入れ換えが可能な複数のストリングを有するグループは第2レベルグループのみである。この第2グループのストリング列(D)-(C)-(B)を上流側で隣接するストリング列(D)-(B)-(C)をで置き換えることにより、隣接するSMOを得る。なお、元の第2レベルグループのストリング列(D)-(C)-(B)に対する辞書の見出し下流側で隣接するストリング列はない。こうして、一つの個体の局所変形個体が全て得られた後、それらの局所変形個体と元の処理中の個体の適応度とを比較し、最も適応度の改良の大きい個体を元の個体の改良個体として選択し、元の個体の代りに使用する。

【0057】その後、他の個体について同様のことをする。こうして適応度の改善が図られたN個の個体1402とそれらの適応度1442を得る。

(3-6-3) 交配プールの生成143

このようにして得られたN個の個体から、適応度が上位にある一定数M個の個体を選択して交配プールを構成する。この数Mは適宜決定する。例えば、 $N/2$ でもよい。この交配プールからランダムに $N/2$ 対の個体1403を選ぶ。この際、交配プール中の各個体はランダムに選択されるため、それぞれが選択される回数は各個体により異なる。併せてそれらの適応度1443を記憶する。交配プールの構成法には、各個体に適応度に応じた選択確率を与え、それを用いて局所探索の結果1402から確率的に選択するという方法もある(参考文献[2])。

【0058】(3-6-4) 交叉操作144

ここでは各対に交叉操作を施し、新たなN個の個体1404を生成し、それらの適応度1444も記憶する。交叉操作では、各対の二つの個体の間で、特定ストリングのSMSコードないしSMOコードの内の特定レベルの部分と交換し、新たな個体対を生成する。そのために、

【0059】まずSMSコードを交換するか、SMOコードを交換するかの交換対象の選択を乱数により行なう。SMSコードの交換が選択されたとき、交換を行うべきストリングを選択する。その選択に当たっては、

(条件1) 個体対の間でそのストリングのSMSコードが異なること、すなわち、交叉により親とは別の対が生成されること

(条件2) 健全な個体の生成を保証するためにそのストリングのマージ先のストリング系列のSMSコードが全

て、その個体対の間で同じである

という二つの条件を満たすストリングを選択する。このような条件を満たすストリングが複数あるときには、それらのストリングの中から乱数により一つのストリングを選択する。選択されたストリングのSMSを処理中の個体対の間で交換する。さらに、選択されたストリングをマージ先とするストリング群があるときには、それらのSMSコードも同時に交換する。

【0060】例えば、図10の(a)の上部に示す個体対の間では、ストリング(B)、(C)共にこれらの個体対の間でSMSコードが異なる。すなわち、ストリング(B)のSMSは6-6と4-7、ストリング(C)のSMSは9-12と5-9である。さらに、ストリング(B)、(C)共、マージ先ストリングは(A)であり、そのSMSコード(なし)が共通である。従って、これらのストリング(B)、(C)はともに上記二つの条件を満たすため、交換の候補となる。ストリング(C)が交換対象に選択され、SMSコードを交換した場合、図(a)の下部に示す新たな個体対が得られる。一方、仮にストリングが(B)が交換対象に選択されたときは、ストリング(E)がこのストリング(B)をマージ先ストリングとするため、このストリング(E)のSMSコードも併せて個体対の間で交換する。

【0061】上記の交換対象の選択において、SMOコードの交換が選択された時は、個体対の間でSMOコードの内の交換すべき特定レベルを選択する。その選択においては、個体対の間でSMOコードの内、そのレベルの部分が異なることを条件とする。この条件を満たすレベルが複数存在するときは乱数を用いて一つのレベルを選択する。SMOコードのうち、こうして選択されたレベルの部分と交換すればよい。例えば、図10(b)では、図の左側に示すように、個体対の間で、元のSMOコードのレベル2部分B-C-DとB-D-Cが相互に異なり、交換可能であり、交換の結果、図の右側に示すSMOコードが得られる。

【0062】以上において、処理中の個体対についてSMSを交換すべきストリングまたはSMO内部分データが見つからないときには、元の個体対をそのまま交叉結果として使用する。このようにして、各個体対にたいして新に個体対が生成される。以上を、他の個体についても行なう。こうして、新にN個の個体1404が決定される。交換操作前のN個の個体の代りに使用される。

【0063】(3-6-5) 突然変異操作145

続いて交叉操作の結果得られたN個の個体の各々に対して突然変異操作を施し、N個の新たな遺伝子表現集合1405とそれらの適応度1445を生成する。突然変異操作は各個体のSMSコード群の各々およびSMOコードの各レベルに対応する部分の値をDAGから決まる制約条件の範囲内で確率的に変化させる。変化させる対象を選択する手順は初期世代の生成あるいは局所探索のと

きの変更箇所の選択手順と同様である。概略的には、まずSMSコードをストリングのレベルの順、ストリングの中ではそれを構成する命令の順に選択する。ついでSMOコードをレベルグループの順に選択する。

【0064】この突然変異操作では、以下の点で局所操作と異なる。変更させる対象を選択する毎に、その対象を突然変異させるか否かを、乱数により確率的に決定する。さらに、この対象を変化させると決定された場合、その対象を変化させる範囲は、局所操作における前述の隣接値に限らない。例えば、変更させる対象がSMSコードである場合で、かつ、それを突然変異をさせると決定した場合、変更後のSMSを、DAGの制限範囲内でランダムに決める。

【0065】さらに、SMOコードのあるレベルの部分を変更する場合も、その部分を取り得る値の範囲の中でランダムに決める。さらに、ある変更対象に対して変更を加えた後、他の対象に変更を加えるときには、先の対象に変更を加えて得られる新たな個体内の当該新たな対象に対して、変更を加える。なお、SMSコードを変更する場合、変更後のSMSコードを、DAGにより許容される範囲内で決めることは、局所探索の場合と同じである。

【0066】たとえば、図11の(a),(b)に突然変異操作の例を示す。(a)はある個体のストリング(C)の第1の命令(10)のSMSコードの値9を突然変異させた例で、ここでは、このSMSを、マージ先ストリング(A)の中の、DAG(図5)で規定される、この命令(10)をマージ可能な範囲((1)―(2)―(3)―(4)―(5)―(6)―(9)の範囲)に属する命令5に対するものに変化させている。さらに、同図(b)はSMOコードのレベル2の部分

を突然変異させる例で、この部分をB-C-DからD-B-Cに変化させている。

【0067】こうして得られた新たな個体の適応度を評価する。

【0068】以上を他の個体にも行なう。こうして、新にN個の個体1405とそれらの適応度1445を得る。

【0069】(3-6-6)世代交替146
最後に世代交代146を行い、新世代の遺伝子表現集合1406を生成する。すなわち、基本的には旧世代の個体を全て新世代で置き換えるが両世代の最良個体は旧世代に属していても残す。そのために突然変異操作145で得られたN個の個体1405の適応度1445と旧世代の最良のN個の個体1402の適応度1442と比較し、適応度が高いN個の個体1406を第2世代の個体として選択する。それらの適応度1446も併せて記憶する。その後、この第2世代の個体1446とその適応度1446を初期世代の個体1401とその適応度1441の代わりに使用して、この第2世代のために局所探索142以降の処理からなる世代処理を繰り返す。この

繰返し前に、各世代の個体適応度の最大値を記憶する。今回生成された新たな世代の個体の適応度の最大値と、このように記憶された過去の世代の適応度の最大値から、世代処理による最良個体の改善が進まなくなったか否かあるいは進みの程度を判別し、進まなくなったときあるいは進みが極度に遅くなったときに、世代処理を終了ないし完了する。世代処理の繰返しを打ち切った後、最新の世代中で最も適応度の高い遺伝子表現を選択し、再度プログラム再生部16により最良アセンブリプログラム3を再生する。

【0070】(変形例) 以上の実施例は以下のように変形することも可能である。

(1) 局所探索142の処理において、各個体の局所変形をその個体に関して可能な全ての局所変形個体を生成する代わりに、その個体の局所変形の過程でその個体より適応度が高い個体を得られた時点で、その個体の局所変形を打ち切る方法もある。この方法によれば各個体の局所変形個体の生成数が減少し、それだけ処理時間が低減できる。勿論、実施例のように、各個体について、生成可能な全ての局所変形個体を生成する方が、より適応度の高い個体を生成する可能性が高まる。

【0071】(2) 局所探索処理自体をスキップすることも可能である。その場合、上記変形例(1)よりさらに処理時間が短縮される。

(3) 局所探索操作、交叉操作、突然変異操作を施す順序は必ずしも上記に限らず、他の順序であって良い。

(4) 最良遺伝子表現探索部14が使用する遺伝的アルゴリズムとしては、実施例で使用した参考文献[3]記載のアルゴリズムに代えて、参考文献[2]記載のアルゴリズムを用いても良い。

(5) 交配プールの構成法は実施例記載の方法に代えて、局所探索処理142で得られたN個の個体から、それらの適応度を用いて確率的に選択する方法が考えられる(参考文献[2])。

【0072】(6) 上記の実施例において、スケジュールを施す命令列はアセンブリプログラムの替りに機械語プログラムであっても良い。この場合は図1の2、3はそれぞれ最良機械語プログラム、最良アセンブリプログラムとなる。

(7) 各個体から再生されたアセンブリプログラムを評価するにあたり、実施例のごとく、元のプログラムを実行予定の実計算機で実際にその再生されたプログラムを実行する代わりに、その計算機の構造と動作を模擬して、この実行時間を推定するシミュレータプログラムを使用してもよい。

(8) 命令スケジューラ11がコンパイラ10の内部に取り込まれていてもよい。

【0073】(実験例) 図12には仮想的なRISCマシンと図4のアセンブリプログラムを対象としておこなった10世代分の試行による適応度の改善状況を個体数

10と20、局所探索あり／なしのケースについて示す。この命令列の最良適応度は69マシンサイクルであることが知られており、それは個体数(NPOPL)20の場合に第4世代(局所探索なし)あるいは第1世代(局所探索あり)で発見された。一般に局所探索ありのほうが改善が早く進む。このように最良個体の探索が少ない世代の試行で行える。図13にそれぞれの場合での、最良適応度を持つ個体への到達履歴を示す。各個体は左から順にストリング(B),(C),(D),(E)のSMSコードを表示して示す。SMOコードは省略する。丸括弧内の数字は各個体の適応度を示す。矢印は遺伝的操作による個体間の遷移を示す。rは交叉操作、mは突然変異操作、lは局所探索操作、sは最良個体の世代送り操作である。

【0074】この図から、本実施例の遺伝子表現と遺伝的操作が適応度の改善に有効に作用していることが判る。局所探索なしの場合の最良個体の遺伝子から命令列を再現すると、(1)-(10)-(2)-(15)-(14)-(3)-(7)-(4)-(13)-(5)-(16)-(8)-(6)-(9)-(11)-(12)-(17)-(18)となる。一方、例題の命令列を文献[1]に記載された従来方式により入れ替えると、(10)-(1)-(7)-(2)-(8)-(3)-(14)-(4)-(16)-(5)-(13)-(6)-(15)-(9)-(11)-(12)-(17)-(18)となり、その適応度は74サイクルで本発明を用いた場合の適応度69サイクルに及ばない。その理由は図5に示すDAG上そこから派生する線(アーク)数の多い命令(10)を命令(1)に先行してスケジュールすることにより、固定的規則の限界を示している。また他の例題では、並べ替え前の命令列の実行サイクル数1621に対し従来方式の並べ替えでは1616サイクル、本発明の並べ替えでは1171サイクルとの結果が得られている。

【0075】

【発明の効果】本発明では、命令列のスケジュールに適した遺伝子表現と遺伝的操作の導入により従来方式より優れた命令列への並べ替えが実現できる。また、本発明では、計算機的设计情報を必要としないで命令列をスケジュールすることが出来るので、計算機特性の変化に自

【図6】

図 6

3.1.1 ストリングベクトル (STV)

(A) : 1-2-3-4-5-6-9-11-12-18-17
 (B) : 7-8
 (C) : 10-15
 (D) : 14-16
 (E) : 13

動的に追従して命令列をスケジュールできる。

【図面の簡単な説明】

【図1】本発明の実施例による命令列スケジュール処理の流れ図。

【図2】図1の処理を実行するための計算機システムの全体構成図。

【図3】図1のストリングマージ図(SMD)生成部(13)の処理の流れ図。

【図4】図1の図1のアセンブリプログラム(2)の例を示す図。

【図5】図4のアセンブリプログラムに対して図1のDAG生成部(12)により得られる命令依存解析図(DAG)を示す図。

【図6】図4のアセンブリプログラムに対して図1のSMD生成部(13)により得られたストリングベクトル(STV)を示す図。

【図7】図4のアセンブリプログラムに対して図1のSMD生成部(13)により得られたストリングマージ図(SMD)の例を示す図。

【図8】図4のアセンブリプログラムに対して図1の初期世代生成処理(141)により得られた遺伝子表現(SMSコードとSMOコードの組)の例を示す図。

【図9】図1の局所探索操作(142)により得られた個体の例を示す図。

【図10】図1の交叉操作(144)により得られた個体の例を示す図。

【図11】図1の突然変異操作により得られた個体の例を示す図。

【図12】図1の最良遺伝子探索処理(14)による10世代にわたる探索の履歴を示す図。

【図13】図1の最良遺伝子探索処理(14)による最良命令列の1つへの到達の履歴を示す図。

【符号の説明】

121…命令解析図(DAG)、131…ストリングマージ図(SMD)、311…ストリングベクトル(STV)、81…ストリングマージ状態(SMS)コード、82…ストリングマージ順序(SMO)コード。

【図7】

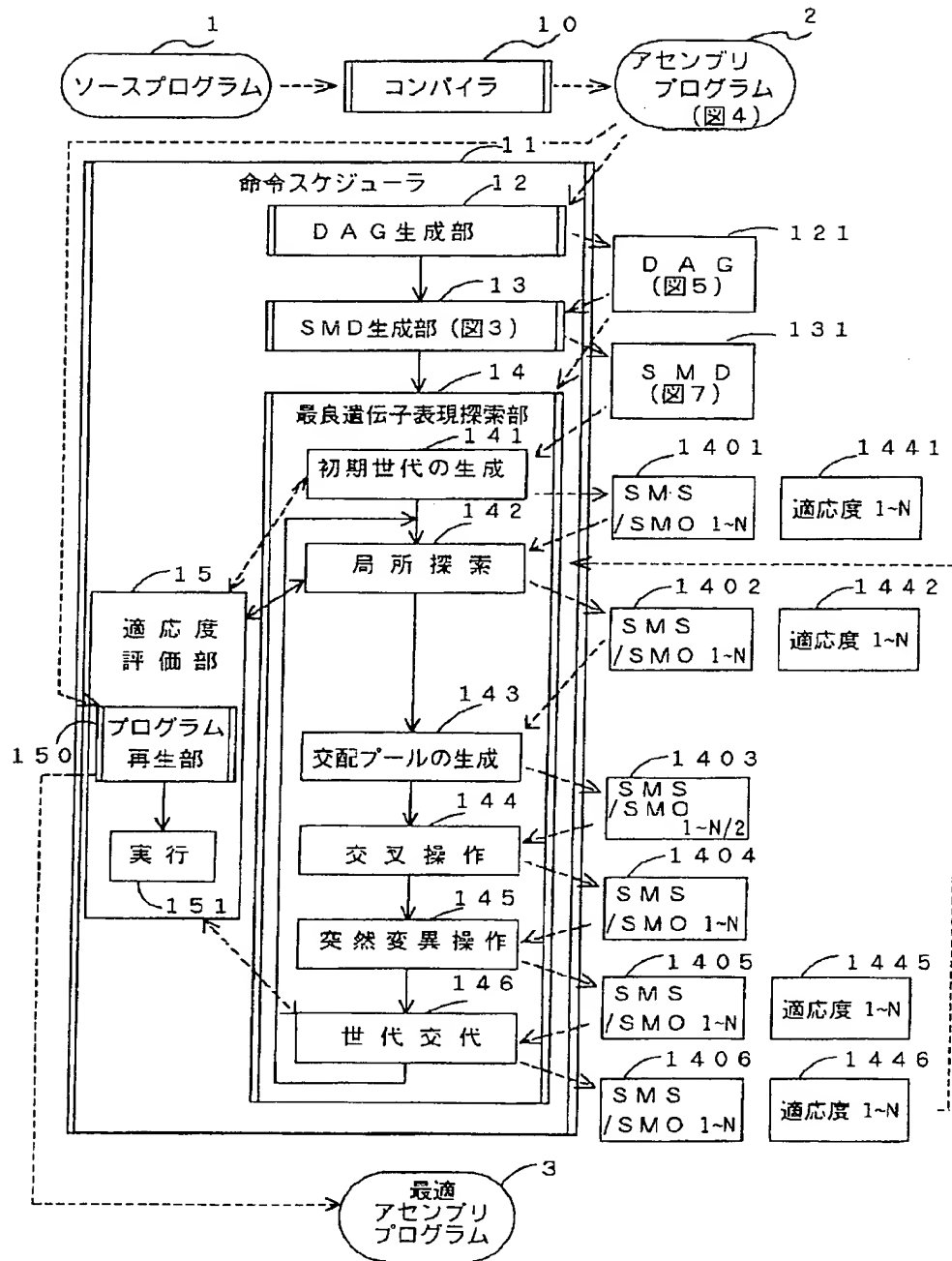
図 7

1.3.1 ストリングマージ図(SMD)

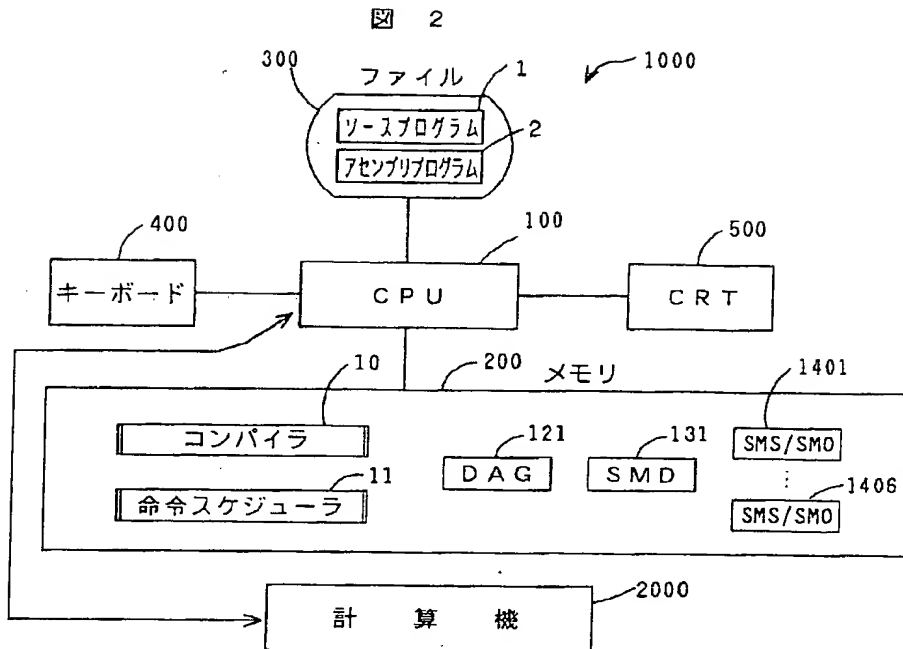
(A) : レベル1 : 1-2-3-4-5-6-8-11-12-18-17
 (B) : レベル2 : 7-8
 (C) : レベル2 : 10-15
 (D) : レベル2 : 14-16
 (E) : レベル3 : 13

【図1】

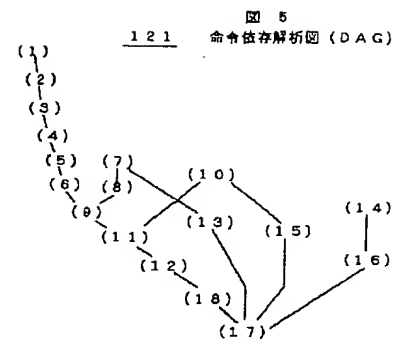
図 1



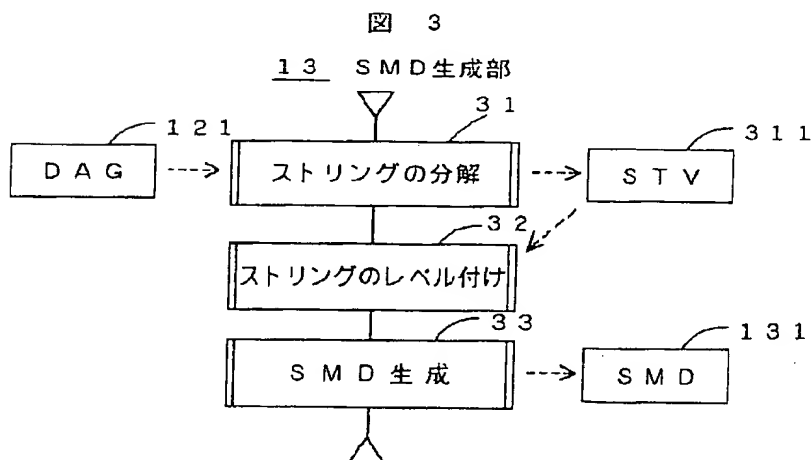
【図2】



【図5】



【図3】



【図12】

図 12

(a) 局所探索なし

NPOPL = 10

gs 0	min	73	max	129	avr	93
1	min	73	max	89	avr	82
2	min	71	max	95	avr	80
3	min	71	max	91	avr	77
4	min	71	max	73	avr	72
5	min	71	max	73	avr	72
6	min	71	max	80	avr	72
7	min	71	max	71	avr	71
8	min	71	max	80	avr	71
9	min	71	max	71	avr	71
10	min	71	max	71	avr	71

NPOPL = 20

gs 0	min	71	max	129	avr	91
1	min	71	max	105	avr	83
2	min	71	max	90	avr	78
3	min	71	max	80	avr	74
4	min	69	max	86	avr	73
5	min	69	max	87	avr	72
6	min	69	max	79	avr	71
7	min	69	max	74	avr	70
8	min	69	max	81	avr	71
9	min	69	max	74	avr	70
10	min	69	max	73	avr	69

(b) 局所探索あり

NPOPL = 10

gs 0	min	73	max	118	avr	87
1	min	71	max	80	avr	74
2	min	71	max	74	avr	73
3	min	71	max	75	avr	72
4	min	71	max	75	avr	72
5	min	71	max	73	avr	72
6	min	71	max	73	avr	71
7	min	71	max	71	avr	71
8	min	71	max	71	avr	71
9	min	71	max	71	avr	71
10	min	71	max	71	avr	71

NPOPL = 20

gs 0	min	71	max	118	avr	85
1	min	69	max	80	avr	73
2	min	69	max	81	avr	71
3	min	69	max	73	avr	70
4	min	69	max	75	avr	70
5	min	69	max	75	avr	69
6	min	69	max	74	avr	69
7	min	69	max	72	avr	69
8	min	69	max	74	avr	69
9	min	69	max	72	avr	69
10	min	69	max	72	avr	69

13

```

gs = 0          gs = 1          gs = 2          gs = 3          gs = 4
25-11-25-18s---> 25-11-25-18
(71)          (71)
      \r      /r
34-4.18-36-9---> 34-11-25-18---> 25-11-25-18
(82)          |m          (72)
      \v      /r      \r
36-56-34-7      26-11-25-18      36-11-25-4
(80)          (76)          |m
      \r      /r      \v
36-56-16-4---> 36-56-34-4---> 36-11-34-4      36-12-25-4
(80)          (75)          (72)          (72)
      \r      /r      \r
26-55-4.11-3---> 12-55-36-7      14-45-36-2---> 14-45-34-2---> 14-45-25-2
(81)          |m          (74)          (72)          (69)
      /r      \v      /m
12-69-36-7      12-45-36-7---> 14-45-36-7
(83)          (74)          ^m
      /r      /r
14-46-16-2      14-5.12-16-2
(79)          (76)
      \r      /m
12-69-36-7---> 14-69-16-2
(83)          ^m

```

r ; 交叉操作
m ; 突然変異操作
s ; 世代おくり操作

```

gs = 0
gs = 1

36-56-34-7 ----> 26-56-34-7
(80)          (75) \ r(SM0)          m(no change) l
                    ----->14-45-16-2----->14-45-16-2 --->14-45-15-2
                                   (69)

14-46-16-2 ----> 14-45-16-2
(79)          (74) /

r ; 交叉操作
m ; 突然变异操作
l ; 局所探索操作

```